

Respectfully submitted,  
BARDMESSER LAW GROUP

/GB/

George S. Bardmesser  
Attorney for Applicants  
Registration No. 44,020

Date: December 10, 2008

1025 Connecticut Avenue, N.W., Suite 1000  
Washington, D.C. 20006  
(202) 293-1191



מדינת ישראל  
משרד המשפטים



רשות הפטנטים



This is to certify that  
annexed hereto is a true  
copy of the documents as  
originally deposited with  
the patent application of  
which particulars are  
specified on the first  
page of the annex.

זאת לתעודה כי רצופים  
בזה העתקים נכונים של  
המסמכים שהופקדו  
לכתחילה עם הבקשה  
לפטנט לפי הפרטים  
הרשומים בעמוד הראשון  
של הנספח.

היום 06-11-2008 This

רשם הפטנטים

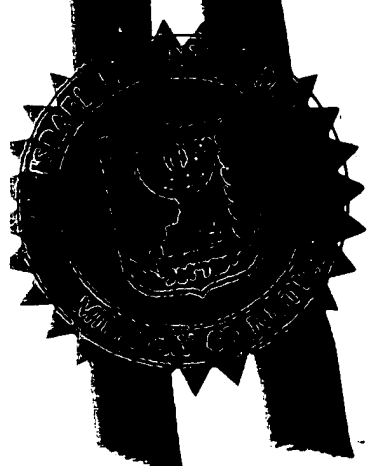
Commissioner of Patents

ד"ר מאיר נועם

*Handwritten signature*

רשם הפטנטים, המדגמים וסימני הפסח

נתאשר  
Certified



אישור הגשת בקשה  
Filing Receipt

Confidential for applicant only

שמור למבקש בלבד

Application No:

145105

מספר בקשה:

Date of application:

23/08/2001

תאריך הבקשה:

METHOD AND SYSTEM FOR PROVIDING A WEB  
SERVICE BY A PLURALITY OF WEB DOMAINS  
THROUGH A SINGLE IP ADDRESS

שיטה ומערכת לאספקת שירות אינטרנט עלידי מספר אתרים  
באמצעות כתובת אינטרנט יחידה

Patentee (s):

בעל (י) הפטנט:

SPHERA CORPORATION  
Customer No: 715361  
29 CRAFTS ST,  
SUITE 360  
NEWTON MA 02458  
U.S.A.

Inventor (s):

ממציא (י) מ:

GREGORY BONDAR  
Customer No: 618611  
NIR BARAM  
Customer No: 618651

Address for service:

המען למסירת הודעות:

LUZZATTO & LUZZATTO  
INDUSTRIAL PARK, OMER,  
P.O.B. 5352,  
BEER-SHEVA 84152

לוצאטו את לוצאטו  
גן תעשייה, עומר  
ת.ד. 5352  
באר שבע 84152

Reference:

13145/01

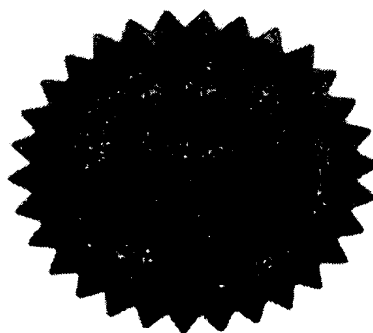
סימוכין:

Invention owner by:

assignment העברה

המצאה מכח:

אישור זה כשהוא חתום בחותמת רשות הפטנטים מהווה אישור הגשת בקשה בלבד ואינו מהווה  
תעודה המעידה על רישום הפטנט. במידה ונמצא שיבוש בקליטת הנתונים כפי שהם מופיעים באישור זה הינכם  
מתבקשים להעיר את הערותיכם תוך עשרה ימים מהיום.



Register of Patents

רשם הפטנטים

as of

05/11/2008

מצב ליום

Ref: 13145/01

שיטה ומערכת לאספקת שירות אינטרנט על ידי מספר אתרים באמצעות כתובת  
אינטרנט יחידה

A METHOD AND SYSTEM FOR PROVIDING A WEB SERVICE BY A  
PLURALITY OF WEB DOMAINS THROUGH A SINGLE IP ADDRESS

**A METHOD AND SYSTEM FOR PROVIDING A WEB SERVICE BY**  
**A PLURALITY OF WEB DOMAINS THROUGH A SINGLE IP**  
**ADDRESS**

5   **Field of the Invention**

The present invention relates to the field of Web hosting. More particularly, the invention relates to a method and system for providing a Web service (HTTP, FTP, POP3, SMTP and other Web services) by a plurality of Web domains through a single IP address

10

**Background of the Invention**

A **Domain** refers to group of Web services provided by or in behalf of an enterprise. Usually it comprises a set of network addresses, each of which provides one or more Web services (HTTP, Telnet, FTP, E-mail, etc.).

15

A **Domain name** is the part of the URL (Uniform Resource Locator) that tells a **domain name server** using the **domain name system** (DNS) whether and where to forward a request for a Web page or Web service. The domain name is mapped to an IP address, which represents a physical  
20   point on the Internet.

**IP-address** (Internet Protocol address) is the address of a computer attached to a TCP/IP network. Every client and server station that can be addressed via the Internet must have an IP address. A specific computer can have more than one IP address associated with it. If a computer has  
5 more than one IP address, these addresses can be associated with different NICs (Network Interface Cards), or several IPs can be associated with one NIC. In some cases, one IP is associated with several computers, using a Load Balancer or firewall. In that case, the external device (load balancer or firewall) translates the external IP to some local IP, and vice-versa.

10

IP addresses are written as four sets of numbers separated by periods; for example, 204.171.64.2. The TCP/IP packet uses 32 bits to contain the IP address, which is made up of a network address (NetID) and host address (HostID). Certain high-order bits identify class types and some numbers  
15 are reserved. On the Internet itself - that is, between the router that move packets from one point to another along the route - only the network part of the address is looked at.

The more bits used for network address the fewer remain for hosts. As the  
20 Internet becomes popular, the IP address resources are exhausted, and consequently the IP addresses become a precious resource.

A few years ago, every domain on the Internet had its own IP address, but currently, due to the rapid depletion of the IP resources, there has been an increased effort to develop technologies for sharing one IP address between a plurality of domains. Actually, the use of one IP address for serving a plurality of Web sites that provide HTTP services was already dealt with in the prior art, and was referred to as **Virtual hosting**.

There are two methods for carrying out virtual hosting: **Name-based virtual hosting** and **IP-based virtual hosting**. In IP-based virtual hosting, one host computer deals with a plurality of IP addresses, each of which corresponds to a domain. In name-based virtual hosting, one IP address is shared by a plurality of domains.

The HTTP/1.1 protocol and a common extension to HTTP/1.0 support name-based virtual hosting, and accordingly Web servers correspond to this protocol. This is implemented by including the Web domain in the HTTP "GET" request. However, in the prior art no solutions to the problem of sharing one IP address between a plurality of domains that provide Web services beyond HTTP (such as FTP and e-mail services) has been presented.

it is an object of the present invention to provide a method and system for providing a web service by a plurality of web domains through a single IP address, which can be implemented for HTTP as well as for FTP, SMTP, POP3 and other Web services.

5

Other objects and advantages of the invention will become apparent as the description proceeds.

### **Summary of the Invention**

- 10 In one aspect, the present invention is directed to a method for providing a Web service by a plurality of Web domains hosted by a computer, through a single IP address, comprising:
  - a) For each of the domains, allocating a server having a unique domain name and the IP address, for providing the service;
  - 15 b) Providing a wrapper, being a software module for intermediating between a client of the service and the servers via a dedicated protocol, and via a standard protocol for communicating with each of the servers;
  - c) Upon receiving a request for connecting the client to the one of the servers in order to provide the service:
- 20 (i) Identifying the target domain name of the request by interacting between the client and the wrapper via the dedicated protocol;



- (ii) Interacting between the wrapper and the server providing the service which is associated with the target domain name by the standard protocol;
- (iii) Establishing a communication channel between the server and  
5 the client; and
- (iv) Allowing the server to provide the service to the client.

The dedicated protocol of may comprise a command for identifying the user and the domain. The domain name may be separated from the user name  
10 by one or more non-acceptable characters in the user name as defined by the protocol.

According to one embodiment of the invention, the Web service is FTP and the domain name phrase is "user%domain" or "domain%user", in which  
15 "user" is the user name, "domain" is the domain name, and "%" is the non-acceptable character(s).

According to another embodiment of the invention, the Web service is POP3 and the domain name phrase is "user%domain" or "domain%user",  
20 in which "user" is the user name, "domain" is the domain name, and "%" is the non-acceptable character(s).

The Web services can be HTTP, FTP, POP3, SMTP, MIRC, Telnet, SSH, Rtelnet, and Shell.

Each of which of the Web domains may refer to a different Virtual  
5 Dedicated Serer.

The computer system may be a Unix-based system, any dialect of Unix, Solaris, Linux (Red Hat, Debian, SuSE, FreeBSD, etc.), AIX, HP/UX, Tru64, and Irix.

10

According to one embodiment of the invention, some or all of the server(s) can be replaced by hard links.

In another aspect, the present invention is directed to a system for  
15 providing a Web service to a client by a plurality of Web domains hosted by a computer, through a single IP address, comprising:

- A server for providing the service, for each of the domains;
- A wrapper, for intermediating between the client and the servers, such that communicating with the client is carried out via a dedicated  
20 protocol, and communicating with the server is carried out via standard protocol.

**Brief Description of the Drawings**

The above and other characteristics and advantages of the invention will be better understood through the following illustrative and non-limitative detailed description of preferred embodiments thereof, with reference to

5 the appended drawings, wherein:

Fig. 1 schematically illustrates a typical FTP session, according to one embodiment of the invention; and

10 Fig. 2 schematically illustrates Web servers hosted by VDS systems that are hosted by one computer system, according to a preferred embodiment of the invention.

**Detailed Description of Preferred Embodiments**

15 Without any loss of generality, the examples herein refer to a Unix-based operating system, such as Solaris, Linux (Red Hat, Debian, SuSE, FreeBSD, etc.), AIX, HP/UX, Tru64, Irix, and so forth.

**Client/server** describes the relationship between two computer programs

20 in which one program, the client, makes a service request from another program, the server, which fulfills the request. Although the client/server idea can be used by programs within a single computer, it is a more

lucrative idea in a network. In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations. The client/server model has become one of the central ideas of network computing. Most business applications being  
5 written today use the client/server model. So does the Internet's main programs which are usually based on the TCP/IP protocol.

A client process referring to an IP address actually communicates with a Web server. A **Web server** is a program that utilizes the client/server  
10 model "serves" requests for its services. Every computer on the Internet that contains a Web site must have a Web server program. On the one hand, a very large Web site may be spread over a number of servers in different geographic locations. On the other hand, one Web server can host a plurality of Web sites.

15

Regarding the Web, a **Web server** is the computer program that serves requested HTML pages or files. A Web client is the requesting program associated with the user. The **Web browser** in the user's computer is a client that requests HTML files from Web servers.

20

In the usual client/server model, one server, sometimes called a **daemon**, is activated and awaits client requests. Typically, multiple client programs

share the services of a common server program. Both client programs and server programs are often part of a larger program or application. Relative to the Internet, your Web browser is a client program that requests services (the sending of Web pages or files) from a Web server (which

5 technically is called a Hypertext Transport Protocol or Hypertext Transfer Protocol server) in another computer somewhere on the Internet. Similarly, your computer with TCP/IP installed allows you to make client requests for files from FTP (File Transfer Protocol) servers in other computers on the Internet.

10

A **Daemon** is a process that awaits incoming requests and then forwards them to other process(es) as appropriate. On the Web, each server has an **HTTPD** (Hypertext Transfer Protocol daemon) that waits in attendance for requests to come in from the rest of the Web.

15

The term **socket** refers herein to a facility of directing data to an application via a TCP/IP network. The combination of the IP address of the station and a port number determines a socket. One computer writes data to a socket in order to send the data to a second computer, and the

20 second computer reads from its socket the data. This can be illustrated as a telephone call. In order to speak with a subscriber, one has to dial its telephone number (in our case the IP address), and then its extension (the

port number). After the connection has been established, one talks to its handset (socket), and the other listens through his handset (socket).

A **well-known port** refers herein to a protocol port that is widely used for a certain type of data on the network. For example, HTTP is typically assigned port 80, FTP transfer is port 21, the POP3 the port number 110, and X-Windows 6000. A **Privileged port** refers herein to a protocol port numbered from 1 through 1023.

Under a Unix-based operating system, the FTP server is a process that holds a communication session through a well-known port. During the communication session, the FTP server performs commands, as defined in RFC 959. **RFC (Request For Comment)** is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties.

The first two commands usually to be performed are the "USER" and the "PASS" commands, which are used for identifying the user name and its password to the FTP server. The user identification is carried out prior to any further commands to the FTP server, such as downloading or uploading files. The FTP server validates the name and the password using the computer's username and password ("/etc/passwd" in a

Unix-based systems), and if they correspond - the user is allowed to access files according to the specific user's permissions on the file system. Of course, more secured mechanisms can be used, and this scheme alone has been described herein for the sake of brevity.

5

**Inetd** (INternET Daemon) is a Unix function that manages many common TCP/IP services. It is activated at startup, waits for various connection requests (FTP, Telnet, etc.) and launches the appropriate server components.

10

According to the prior art, an FTP server or any other Internet server can be activated in two modes:

- The "Inetd" mode, in which a single process (the Inetd daemon) serves a plurality of network services. The daemon "listens" on specific ports, waiting for requests for connection. When a request for connection arrives, it creates the service process (according to the well-known port), and allows it to handle the communication session.

20

The benefit of this approach is that the system resources are saved since there is a single process that listens on all the ports instead of a plurality of processes.

- The "Stand-alone" mode, in which the relevant process is created once (for example, when the system boots), and the process handles the connections. This mode suits services that typically are active all the time and therefore starting and terminating a process every time a connection arrives results in an unnecessary overhead.

Usually, HTTP server operates in the Stand-alone mode, while the FTP server, POP3 server, and SMTP server operate in the "Inetd" mode.

#### The problem of name-based FTP and e-mail

In the original design of the FTP as described in RFC 959, the FTP was directed to serving one domain, associated with one IP address, unlike the HTTP service in which the commands contain the object domain. Hence, if, for example, two domains *ftp.aaa.com* and *ftp.bbb.com* reside on the same hosting computer and share the same IP address, then the users *xxx@ftp.aaa.com* and *xxx@ftp.bbb.com* are the same. Moreover, if a user tries to access "*ftp.aaa.com/pub/*", he will reach the same directory as "*ftp.bbb.com/pub/*".

Those skilled in the art might determine a partial solution to this issue, by defining specific sub-directories for *ftp.aaa.com* and for *ftp.bbb.com*, and



prevent the users of each domain accessing the other directory. However, there are several problems inherent in such an approach:

- The users would know that there is another directory, but they would not be able to access it. This might be a drawback in Web hosting companies, as it is preferable that each domain be unaware of the existence of other domains residing on the same hosting computer.
- Only one user with a specific name ("xxx" in the above example) can exist on a hosting computer, and therefore common names (like "webmaster", "jobs", "info" etc.) cannot be allocated to a plurality of domains. It should be noted that in order to support this solution, a system should keep the users of each domain in a separate location. For example, the VDS technology, as described in the copending patent application filed under attorney's docket 13010/01, is an example of such a solution.
- The same instance of the FTP server is used for all the domains hosted by a hosting computer. Therefore, there is no trivial way to calculate the resources consumed by a domain, as the process resources are shared.

- By changing the permissions of a file, the file may become accessible to other users. This situation may occur to "rookie" users.

### The solution

5    Actually, the root of the problem described herein is the protocol, which didn't take in consideration of using the same computer for providing FTP services to a plurality of domains. The same problem sustains to mail services as well. Due to the limitation in the protocol, the services providers are limited as well.

10

    The problem of the "missing" domain applies only to POP3. In SMTP the domain is passed as well, and therefore there is no need to add the domain as part of the user name, as for FTP. However, the wrapper is still required for handling the requests by the correct server, i.e., the server of  
15    the appropriate VDS.

    According to the solution disclosed herein, an intermediary between the client and the servers is added in the communication chain. The role of the intermediary is to identify in a request for service the domain of the  
20    request, and to direct the request to this server in a standard form.

Thus, the intermediary interacts with the client as it was the service provider, and with the service provider as it was the client.

Actually, after the domain is identified, the client can interact directly  
5 with the service provider associated with the appealed domain, i.e. without any intermediation.

#### Independent user naming system

According to the prior art, a hosting computer that hosts the domains  
10 *domai1.com*, *domain2.com*, and so forth, cannot have two or more users with the name *sysadmin*, for example.

According to one embodiment of the invention, in order to enable an independent user naming system for each domain hosted on a hosting  
15 computer system, the domain name is embedded in the user name. Referring to the previous example, instead of identifying himself by the user name, i.e. *xxx*, a user identifies himself as *xxx%aaa.com*, *xxx%bbb.com*, and so forth.

20 Practically, on issuing a USER command according to the standard FTP protocol, the user identifies himself by a name wherein the domain is

embedded, which according to this example is *USERNAM%DOMAN*,  
i.e. the user name followed by the character "%" and the domain name.

It should be noted that selecting a character that is legal for username  
5 (such as "\_") might yield to uncertainty. For example, "a\_b\_c" can be  
interpreted as "a%b\_c", or "a\_b%c".

Of course, this syntax does not suit the standard protocol of the FTP as  
defined in RFC 959, and therefore an additional step should be carried out,  
10 as described herein.

#### Hosting a plurality of domains by one computer system

According to one embodiment of the invention, the FTP process is modified  
in order to support this format. This solution has some drawbacks:

15

- Typically, computer programs have security holes. However, for  
common programs there are patches frequently, and these updates can  
not be applied to the enhanced program, thus the enhancing company  
should run this line of updates itself. Most companies would not like to  
20 have a product unrelated to their core business, and therefore would  
like to leave the FTP server to companies specializing in that field.

- Some of the common FTP servers are "open source", and therefore customers would like to use this server, and not the custom made one from the specific company.

5 According to another embodiment of the invention, a component that handles the communication part is added to the communication chain. This component handles the connection to a 3rd party's FTP server (i.e. the owner of the computer can use any FTP server – open source, commercial, or even homemade) when the user is identified. For the sake  
10 of brevity, it is assumed that this component is a process. This process is hereinafter called **FTP-wrapper** or **wrapper**.

When a connection to the FTP port arrives to the hosting computer, the daemon creates an FTP-wrapper process. The FTP-wrapper "negotiates"  
15 with the client as an FTP server, i.e., receives commands from the client and relays answers to the client as the FTP server. The negotiation is carried out until the client issues the "USER" command.

It should be noted that the wrapper might support only a subset of the  
20 FTP commands, as most commands are not supported before the user logged into the FTP server. Once the FTP-wrapper received the "USER"

command, it confirms that it includes the user name and a valid domain name, and starts an FTP server for that domain.

It should be noted that in order to achieve better performance, the  
5 wrapper might be integrated into the daemon, so the creation of the wrapper process for each connection is unnecessary.

Once the original FTP server is created, it expects to have a communication from the beginning, including the "USER" command, and  
10 any prior commands that the user issued.

According to one embodiment of the invention, the wrapper process is kept active for the entire FTP session, thus acting as an interface between the client and the FTP server - every request for service is passed to the FTP  
15 server, and any result from the server is passed to the client.

This solution has several drawbacks:

- The number of processes handling FTP commands is twice the number of processes handling a regular operation, which consumes system  
20 resources and therefore degrades the performance.

- A byte transformed between the client and the server is actually transferred twice – from the client to the wrapper and then to the real server, and vice versa. This reduces performance as well.

5    **A Shared library** (or DLL in Windows operating system) is a collection of functions, usually related to some specific subject, that are kept in an independent file accessible to all the processes at run-time.

According to a preferred embodiment of the invention, the solution to the  
10    above-mentioned problem is based on that fact that most Unix-based processes are linked to shared libraries. The benefits of this approach are:

- Each process is smaller, as the common functions are not part of the process itself. This requires less disk space for storing the program, and less disk space for distributing it.
- 15    • Since the operating system can load a single copy of the library into memory and use it for several processes, the amount of memory required for several processes that use shared libraries is less than the amount needed for the same processes when executed without shared libraries.
- 20    • In order to include new features in an existing function, correct bugs or correct security holes, new versions of the functions are developed. Upon installing a new version of a function in a shared library,

processes being loaded at run time refer to the new version of the function, while other processes do not.

According to one embodiment of the invention, a new shared library that  
5 replaces the relevant functions of the original socket-library is activated, but the original socket-library is retained in order to be used later. A function of the new shared library performs some additional operations (which are not a part of the original function), and then activates the original library's function with the same name.

10

This mechanism is well-known in the art, and is referred to as **hooking**. A **hook** is a place (and usually an interface) provided in packaged code that allows a programmer to insert customized programming, such as additional features.

15

According to one embodiment of the invention, the hooking is carried out as follows:

- A buffer is provided to each socket, for retaining temporarily the information received from the client.
- 20 - During the operation, if the buffer is not empty, "read" commands read the data from this buffer, and if the buffer is empty, then the "read" command retrieves the data from the socket.



- Any "write" command ignores the data until the "read" buffer is empty. After that, all the information is transferred to the socket. Whenever the process using the library performs a "write" command, the library checks the status of the internal buffer. If the internal buffer comprises any information, the information passed to the "write" command is ignored and a "success" status is returned to the caller, as this information was already handled by the wrapper. If no more data is present in the internal buffer, the information is passed to the normal sockets library.

10

According to a preferred embodiment of the invention, a special version of the socket library functions is provided as a shared library. The wrapper process refers to this library. The special version of the socket library writes data sent to or received from the socket into a buffer. Once the "USER" command is received, the FTP-wrapper process creates the FTP process. From this point on, the FTP server communicates directly with the client without the intermediation of the FTP-wrapper, i.e., the original functions are called, rather than the functions of the special version of the library. This results in a slight overhead.

20

#### An FTP session

Fig. 1 schematically illustrates a typical FTP session, according to one embodiment of the invention.

- At step 1, a client connects to an FTP port of a hosting computer. The  
5 Inetd receives the request, opens a port (i.e., communication channel with the client).
- At step 2, carried out once the communication channel with the client has been established, any received FTP command except the USER command is stored in a buffer rather than processed (but a suitable  
10 reply/acknowledgment is sent to the client, so that the client will not hang up), until a USER command arrives.
- At step 3, the "USER xxx%aaa.com" command is received. Only the command that is relevant to the FTP process is stored in said buffer, i.e. - "USER xxx".
- 15 - At step 4, an FTP process for the *aaa.com* domain is created.
- At step 5, if the wrapper is a process (rather than a function), the execution of the wrapper is terminated. It should be noted that from steps 1 to 5 the session has been carried out between the client and the Inetd or by the wrapper, and not between the client and the FTP  
20 server, as in the prior art.
- At step 6, the FTP process receives the commands stored in said buffer. Actually, the FTP server communicates with the wrapper (or with the

information that the wrapper retained in the buffer) instead of with the client, but the FTP server is not "aware" of this fact. From this point on, the client communicates with the FTP server, as in the prior art.

- At step 7, the PASS command is sent by the client to the FTP server in order to be validated.
- At step 8, after the password has been validated by the FTP server, the client can send file-related commands such as download, upload, delete, and so forth, to be performed by the FTP server. It should be noted that the verification of the password is handled by the FTP server, and not by the wrapper. Therefore, if the user prefers to install special authentication modules for its FTP server, or to perform special actions on a failed login – these operations will be performed anyway.

#### Applying the invention to other Web services

- 15 With a **Web mail** service, one can view and respond to his new e-mails from any computer with an Internet connection. The e-mail messages remain on the mail servers until the user checks his mail from his home computer again.
- 20 With **POP** (Post Office Protocol), mail is delivered to a server accessible through the Web, and a personal computer user periodically connects to

the server and downloads all of the pending mail to the "client" machine. Thereafter, all mail processing is local to the client's machine.

SMTP (Simple Mail Transfer Protocol) is the standard e-mail protocol on  
5 the Internet. SMTP servers route SMTP messages throughout the Internet to a mail server, such as **POP3** or **IMAP4**, which provides a message store for incoming mail.

Like the FTP protocol, Web mail protocols restrict the provision of services  
10 to a user by authenticating the user, which typically is carried out by the user name and a password. Since a user is usually associated with an account on the hosting computer, the provision of such services can be implemented in the same way as FTP.

15 **HTTP** (HyperText Transport Protocol) is the communications protocol used to connect to servers on the World Wide Web. Its primary function is to establish a connection with a Web server and transmit HTML pages to the client browser. Addresses of Web sites begin with an "http://" prefix.

20 For HTTP, there is a well-known solution called "Virtual hosting". Using this technique, the HTTP server can detect the domain that the user wishes to access (from a special line in the command), and therefore to

access a different directory tree for each domain. However, this solution implies sharing the same Web server between the domains. Hence, this solution does not suite a Web hosting company, which has an interest to limit the resources that each domain receives, and probably charges each  
5 account according to its actual use.

According to one embodiment of the invention, a plurality of instances of a Web server can be executed simultaneously, such that each domain (or a group of domains) uses its own Web server instance. However, since only  
10 one Web server can use a certain port on a computer at a given moment, only one Web server can be active at that moment.

This obstacle can be removed by piping the data from the original socket to an "internal" socket, which actually can be accessed by a plurality of Web  
15 servers at a given moment.

Hence, the invention may be implied to HTTP services as well as to FTP and mail. Implementing the virtual hosting methods used in the prior art, i.e. the virtual hosting ability that is provided by a Web server, all the  
20 domains are served by a single Web server. However, according to the invention, several Web servers can run simultaneously, each one serving

some of the domains. Therefore, a WHP can achieve better performance of the hosted Web sites.

Of course, this solution enables limiting the number of connections that a specific Web server accepts, or limiting its total resource usage using some Quality of Service mechanism. Quality of Service is the continuous transmission of high-bandwidth video and multimedia information.

It should be noted that receiving the connection, reading the requested domain and handing the socket with the information to the relevant Web server, is transparent to the server.

The above examples and description have of course been provided only for the purpose of illustration, and the invention can be implemented to any Web service including MIRC, Telnet, SSH, Rtelnet, and Shell.

The secured version of these services, as well as any other secured service, can be implemented in a similar approach. However, it is not straightforward, as the secured services usually use a different encryption key for each domain. This key is used to prevent unauthorized computers from accessing the information, as well as to authenticate the server that

the information is sent to. The solutions for implementing the same approach for the secured services might be:

- To use the same key for all the domains on each server (which weakens the solution, as all the domains on the server can decrypt the messages, even if intended for other domains on the computer);
- To modify the protocol to contain some information about the target domain in clear (i.e. not encrypted) text. However, this would require modifying the client application that uses this protocol, and there are a variety of such applications on the market. Therefore, this solution is problematic and suitable only cases in which the client software can also be dictated.
- To try to decode the message using all the domains keys. Once the message is decrypted, the wrapper can pass the message to the target domain (which can be determined by the key upon which the decryption of the message is carried out). It should be noted that according to this solution, there is no need for extension of the user name (as in the present invention), as the target domain can be determined by the encryption key. The drawback of this solution is the fact that decrypting the first message of every connection requires time and computer resources.

#### A name-based VDS

According to the copending patent application filed under attorney's docket 13010/01, an emulation of a computer system in which a remote client can access its system utilities and programs is referred to as a **Virtual Dedicated Server (VDS)**. According to said application, a plurality of  
5 VDS instances can be executed simultaneously on one hosting computer system, each referring to a different directory tree as its root directory.

Using the VDS technology, the FTP server can use a different "/etc/passwd" for each domain, thus resulting in a different set of allowed  
10 users for each domain (including the same user name in several domains).

Using the technology described herein, a plurality of VDSes can use a single IP address for the Web services provided by the VDSes.

15 Fig. 2 schematically illustrates Web servers hosted by VDS systems that are hosted by one computer system, according to a preferred embodiment of the invention.

The hosting computer 10, hosts the VDS systems 60 and 70. VDS 60 is  
20 hosting the FTP server 11, and the POP3 server 12. VDS 70 is hosting the POP3 server 21, FTP server 22 and HTTP server 23. Daemon 40 "listens" to the well-known FTP port, HTTP port, and POP3 port. Whenever the



daemon 40 indicates a request for connection from a client 30, the wrapper 50 "negotiates" with the client 30 as the appropriate Web server until the domain is indicated. Then, the wrapper negotiates with the appropriate Web server 11, 12, 21, 22 or 23 as it was the client. At the last stage, a  
5 connection between the appropriate Web server and the client is established, until the communication session ends.

The "appropriate server" is indicated by two parameters: the port, which indicates the type of the server (FTP, HTTP, etc.), and the domain that  
10 indicates the appropriate VDS.

As described in the copending patent application filed under attorney's docket 13010/01, the performance of the system can be improved using hard links.

15

The above examples and description have of course been provided only for the purpose of illustration, and are not intended to limit the invention in any way. As will be appreciated by the skilled person, the invention can be carried out in a great variety of ways, employing more than one technique  
20 from those described above, all without exceeding the scope of the invention.

CLAIMS

1. A method for providing a Web service by a plurality of Web domains hosted by a computer, through a single IP address, comprising:
  - a) For each of said domains, allocating a server having a unique  
5 domain name and said IP address, for providing said service;
  - b) Providing a wrapper, being a software module for intermediating between a client of said service and said servers via a dedicated protocol, and via a standard protocol for communicating with each of said servers;
  - 10 c) Upon receiving a request for connecting said client to the one of said servers in order to provide said service:
    - (i) Identifying the target domain name of said request by interacting between said client and said wrapper via said dedicated protocol;
    - 15 (ii) Interacting between said wrapper and the server providing said service which is associated with said target domain name by said standard protocol;
    - (iii) Establishing a communication channel between said server and said client; and
    - 20 (iv) Allowing said server to provide said service to said client.

2. A method according to claim 1, wherein said dedicated protocol comprises a command for identifying the user and the domain.

3. A method according to claim 2, wherein the domain name is separated  
5 from the user name by one or more non-acceptable characters in the user name as defined by said protocol.

4. A method according to claim 3, wherein said Web service is FTP and the domain name phrase is "user%domain" or "domain%user", in which  
10 "user" is the user name, "domain" is the domain name, and "%" is the non-acceptable character(s).

5. A method according to claim 3, wherein said Web service is POP3 and the domain name phrase is "user%domain" or "domain%user", in which  
15 "user" is the user name, "domain" is the domain name, and "%" is the non-acceptable character(s).

6. A method according to claim 1, wherein said Web services are chosen from among HTTP, FTP, POP3, SMTP, MIRC, Telnet, SSH, Rtelnet,  
20 and Shell.

7. A method according to anyone of claims 1 to 6, wherein each of which of said Web domains refer to a different Virtual Dedicated Server.
8. A method according to any one of claims 1 to 7, wherein said computer  
5 system is a Unix-based system, any dialect of Unix, Solaris, Linux (Red Hat, Debian, SuSE, FreeBSD, etc.), AIX, HP/UX, Tru64, and Irix.
9. A method according to claim 1, wherein some or all of said server(s) are replaced by hard links.
- 10
10. A system for providing a Web service to a client by a plurality of Web domains hosted by a computer, through a single IP address, comprising:
- A server for providing said service, for each of said domains;
  - 15 - A wrapper, for intermediating between said client and said servers, such that communicating with said client is carried out via a dedicated protocol, and communicating with said server is carried out via standard protocol.
- 20 11. A method for providing one or more Web services to one or more Web domains, substantially as described and illustrated.

12. A system for providing one or more Web services to one or more Web domains, substantially as described and illustrated.

לוצאטו און לוצאטו  
LUZZATTO & LUZZATTO  
ע"מ

	Client		Inetd / wrapper		FTP server
1.	Connect to the FTP port	→	Get the request. Open a port, and get commands.		
2.	Various commands	→	Keep the information in the buffer.		
3.	"USER xxx%aaa.com"	→	Keep only "USER xxx" in the buffer.		
4.			Create an FTP process, for the "xxx" domain.	→	
5.			Finish execution if "wrapper" is an independent process.		
6.					Get commands from the internal buffer. (which seem like arriving from the network).
7.	"PASS" command	→			Verify the password.
8.	Command to download or upload files	→			Handle the commands.

**Fig. 1**

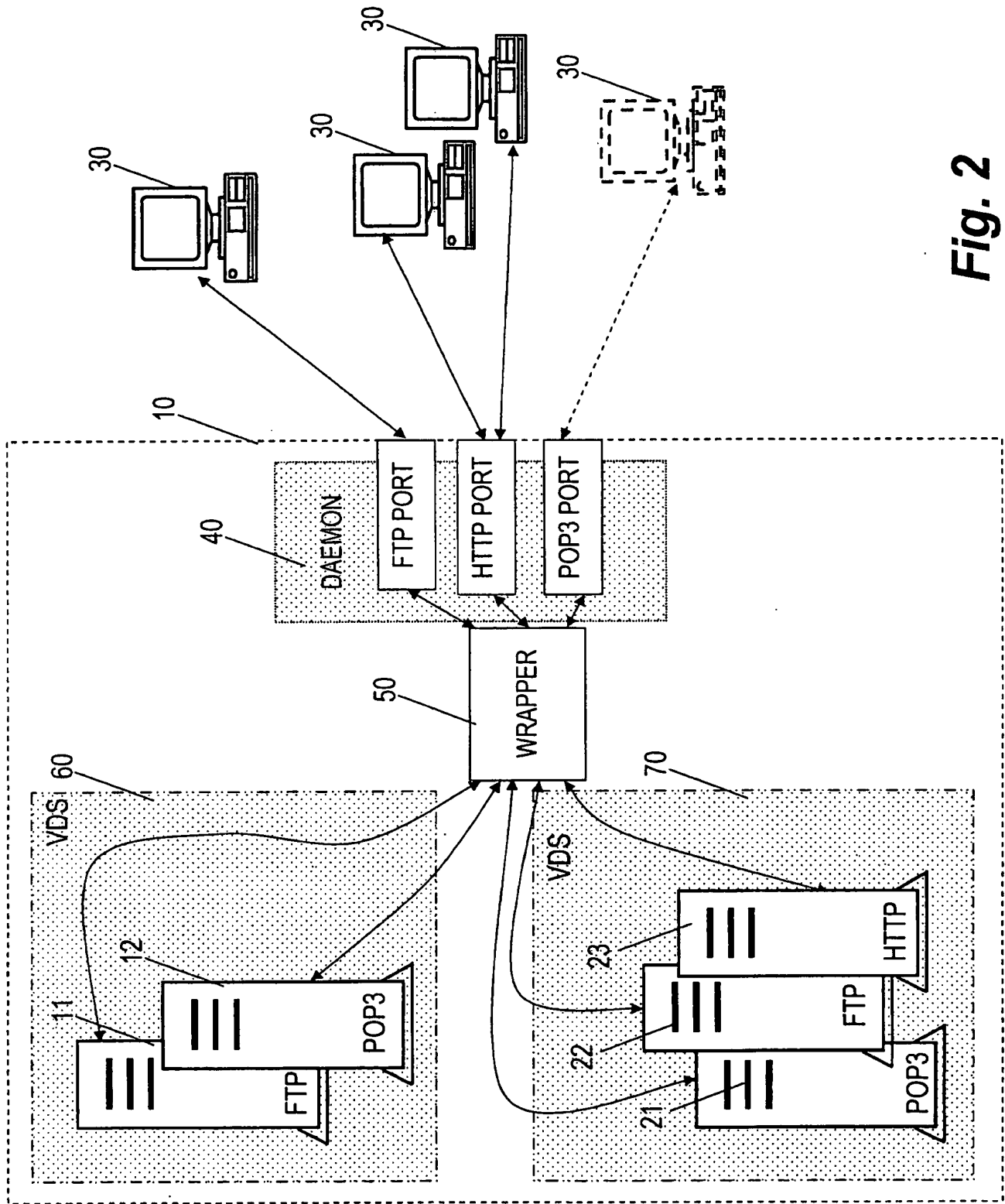
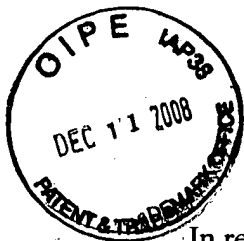


Fig. 2



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

BONDAR *et al.*

Appl. No.: 10/784,570

Filed: February 23, 2004

For: **A METHOD AND SYSTEM FOR  
PROVIDING A WEB SERVICE BY  
A PLURALITY OF WEB  
DOMAINS THROUGH A SINGLE  
IP ADDRESS**

Confirmation No.: 1077

Art Unit: 2141

Examiner: NGUYEN, DUSTIN

Atty. Docket: 2230.1040000

**Claim For Priority Under 35 U.S.C. § 119(a)-(d)  
In Utility Application**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

Sir:

Priority under 35 U.S.C. § 119(a)-(d) is hereby claimed to the following priority document, filed in a foreign country within twelve (12) months prior to the filing of the above-referenced United States utility patent application:

Country	Priority Document Appl. No	Filing Date
ISRAEL	145105	August 23, 2001

A certified copy of each listed priority document is submitted herewith. Prompt acknowledgment of this claim and submission is respectfully requested.